An X-Force™ White Paper

INTERNET
SECURITY
SYSTEMS™

Risk Exposure Through Instant Messaging And Peer-To-Peer (P2P) Networks

April 2002

6303 Barfield Road • Atlanta, GA 30328 Tel: 404.236.2600 • Fax: 404.236.2626

Introduction

The popularity of Instant Messaging and peer-to-peer networking technologies has risen dramatically in recent years. These services are becoming more prolific not only because of the convenience of the instant communication they provide, but the increased deployment of broadband has led to a rise in the availability of movies, music, and other media for download. As these services become increasingly popular, an increased risk emerges as well. Personal users of peer to peer services consider them to be innocuous, and the risks associated with installation and use are not well understood by IT organizations and end users. Many of these technologies were not designed to carry sensitive data in a corporate environment, and therefore do not have encryption or other security features. Clients for certain chat networks are designed to help evade filtering and policy control.

Users of these services are unknowingly putting information about themselves or their companies at risk. The three major instant messaging vendors (AOL, Yahoo!, and Microsoft) have had issues with privacy violations and well-publicized security holes. There have been numerous security risks inherent in peer-to-peer clients such as Morpheus, KaZaA, and the various Gnutella clients, which can also be harnessed to distribute worms and malicious code. This document details the dangers of using these services. It also explores the potential of these clients for misuse and what steps can be taken to minimize their inherent risks.

Architecture

Instant Messaging uses a client-server architecture to send and deliver messages and content. In this model, the client is installed on a particular computer by an end user and is the interface that he or she uses to communicate with others. The server manages and relays all client/user communication and is maintained by a service provider such as AOL or Microsoft. In this model, the server is not only responsible for delivering messages to the intended recipients, but is also responsible for authenticating users and verifying their online status. Clients are not directly connected to each other by default; however this is a feature that can be enabled on some clients.

Peer-to peer networks such as Morpheus and Gnutella use an architecture in which all parties are considered equals. There is no central server and authority in this model, and peers can associate and connect with each other directly. All peers can deliver messages and files without the aid of a centralized server. Additionally, these peers can have multiple connections to other peers at any given time, and can communicate and send files to many different users simultaneously.

AIM (AOL Instant Messenger)
http://www.aim.com/
43.6 million users

ICQ
http://www.icq.com
7.2 million users

.NET Messenger
http://messenger.msn.com
18.5 million users

Yahoo! Messenger
http://messenger.yahoo.com/
11.9 million users

Source: *Digital Media Report*, Jupiter Media Metrix, November 2001

AIM

SECURITY CONCERNS AND AIM

AIM has had several security-related issues, the most recent being a buffer overflow in the game request parsing engine, which was reported on January 2, 2002 by Internet Security Systems' ***X-Force™*** knowledge services organization. This means a certain type of specially crafted game request could be made to an AIM user, causing an area in memory to be overwritten with arbitrary data supplied by an attacker. This data could be coerced into executing on the remote user's computer, thus enabling an attacker to take control. AOL has patched this bug, but this isn't the first bug of its kind to affect AIM. Threats such as these are ever present as the code for the clients become more and more complex. As a result, more and more programming errors will be made.

AIM PROTOCOL

The AIM protocol has been published through various sources, including AOL to aid in development of Linux (or other unsupported) clients.

All AIM commands, messages, and requests are sent through one of many central servers on their system. There are two types of servers on the AIM network: the OSCAR server, which stands for Open System for Communication in Realtime, and the BOS, or Basic OSCAR Service servers. While the OSCAR server is responsible for authorizing clients, there are many BOS servers responsible for handling various features of the AIM service. For example, there are BOS servers dedicated to instant messaging and others dedicated to the locate tool used for finding information about buddy nicknames. OSCAR consists of several IP addresses that resolve in a round-robin manner to login.oscar.aol.com. The BOS servers are not listed and may change at any time. There may also be multiple special purpose servers for a single purpose.

By default, all communication between the server and client happens via port 5190. However, the client is capable of connecting to an OSCAR/BOS server on any port by changing the server port number on the Connection Preferences screen. Therefore, blocking port 5190 on a firewall or other access control device might not prevent AIM clients from connecting.

The AIM protocol consists of a layer above TCP/IP known as FLAP. FLAP is a low-level protocol used on TCP connections between the client and server. A FLAP header makes up the first 6 bytes in any AIM message. For details on the exact construction of FLAP packets, consult the documentation at: http://boxnet.dhs.org/mirror/aol_protocol/OSCAR_Documentation.html

An AIM session begins with a sign on process. The first FLAP packets sent to OSCAR contain an encrypted password, as well as the user's AOL screen name. The password is encrypted using weak XOR and is easily decrypted. The packet sniffing software 'dsniff' (available at http://www.monkey.org/~dugsong/dsniff/) is able to decipher AIM passwords on the fly.

Once an AIM user has signed on with OSCAR, the OSCAR server issues a cookie that enables the user to sign on to each of the special purpose BOS servers without needing to authenticate again. This feature streamlines the protocol somewhat.

After the session is established, the user remains connected to OSCAR and the BOS servers they are using for as long as the session is required. Once the user signs off OSCAR, the cookie becomes invalid for use on the BOS servers and they are disconnected from each.

AIM also has the ability to work with proxy servers using the following protocols:

SOCKS 4
SOCKS 5
HTTP
HTTPS

AIM CAPABILITIES
AIM has the following main features:

- Instant messaging
- IM Images (transfer of inline images in Instant message conversations)
- Voice chat
- Game requests
- File transfers
- File sharing

Instant Messaging
Instant messaging is simply the passing of HTML-encoded clear text messages from one user to another, via a central BOS server. The message is not encrypted and is always routed over the Internet.

IM Images
IM images are sent via a direct connection with another peer. A request is sent to the BOS server and is relayed to the target user.

The request packet for direct connection contains the TCP/IP address and port information of the requester. If accepted, the target of the request listens for an incoming request on port 4443 and a conversation begins between peers. These direct connections reveal the IP address of each participant.

Figure 1. Initiating IM Images Connection in AIM

Figure 2. Accepting IM Images in AIM

Voice Chat
A direct connection must also be used for Voice Chat. Like in IM Images, data is transferred directly between peers. In this case it is sound data instead of images.

Game Requests
Game requests are simply requests for remote users to execute certain external programs, usually games. During game requests, no direct connection is made with peers via AIM. If the external application or game requires a direct connection, one may be set up. This is beyond the scope of AIM.

File Transfers
File transfers are very similar to image transfers in that a direct connection is established. However, once a file transfer is complete the direct connection is closed.

As in IM Images, a BOS server relays a request packet to another user. When the recipient accepts the connection, a TCP port is opened to accept the incoming file. By default, this TCP port is 5190; however it is possible for the user initiating the file transfer to select any available port by nominating it in the File Transfer options dialog.

The default security option is to allow file transfers from all users after displaying an accept file dialog box. These options can be configured in File Transfer Options in Preferences.

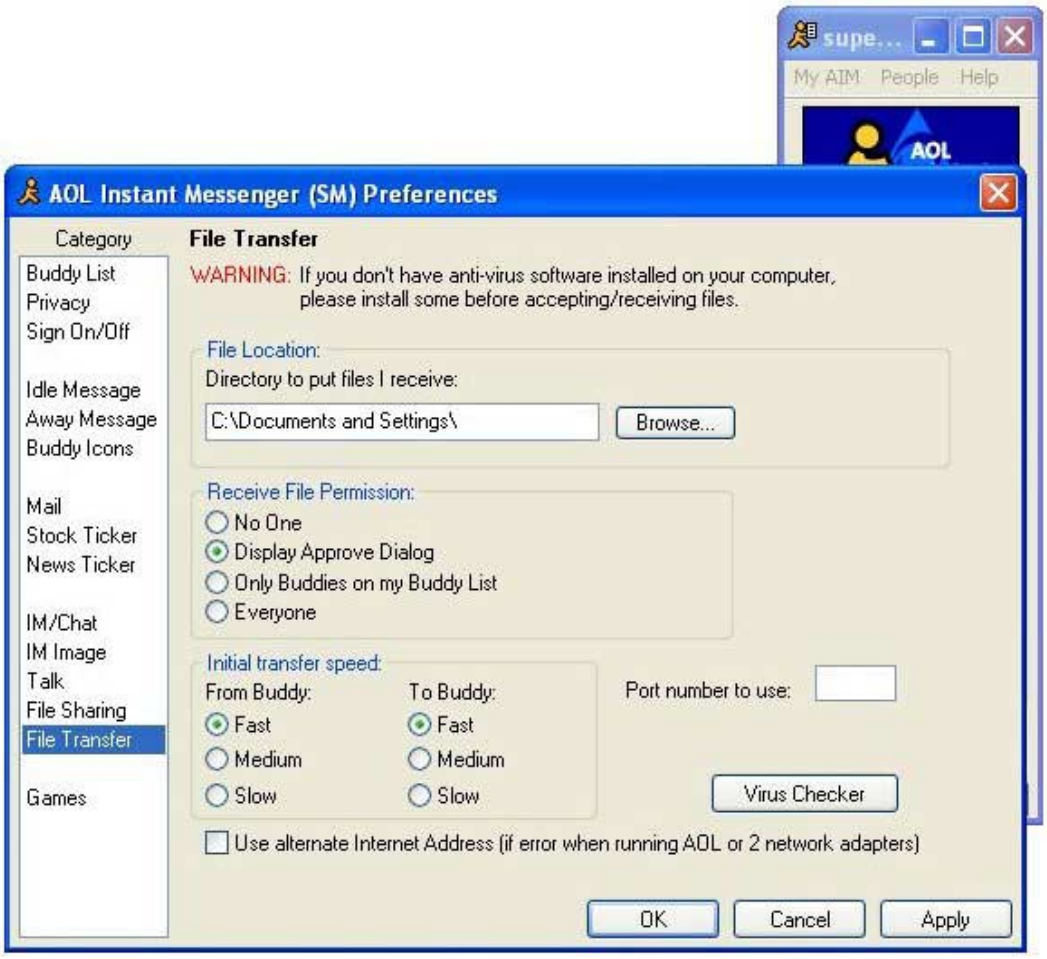

Figure 3. AIM Preferences and File Transfer Permissions

File Sharing

File sharing in AIM is a method that allows a user to browse a selected directory structure and to download files. File sharing is optional and must be enabled before any sharing can take place.

The connection method for file sharing is the same as for a regular file transfer. The initiator sends a request packet to the target via the BOS server. After the target client accepts the request, the initiator begins listening on port 5190 (this can be changed in the File Sharing options dialog) and the target sends the file list information. All file transfers are carried out over the same connection.

If the shared directory is empty, the connection attempt will fail.

AIM SECURITY RISKS

Infected Files: Trojans and Viruses

Any user with an AIM account has the ability to send another user a malicious or infected file.

Misconfigured File sharing

AIM's file sharing feature is configurable, and can be set up to mistakenly share directories in which some or all of the information is sensitive or confidential. An anonymous AIM user might

stumble upon sensitive data such as company documents, system passwords, personal information, etc.

Unencrypted Communication
AIM's protocol stack does not include a secure layer. There is no encryption of any communication sent or received via AIM. Users may send sensitive data via messages or file transfers. As noted before, messages (and file transfers in most cases) are routed via the Internet. If documents are not strongly encrypted, they should not be sent via AIM unless they are already a matter of public record. Users should send information over AIM only with the assumption that a larger audience will have access to this data as well.

Copyright Infringement
Many file transfers completed through AIM violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over AIM eliminates the file size restrictions of email, and, if the recipient is known, AIM is an easier solution for file transfers compared to File Transfer Protocol (FTP).

Social Engineering
Some malicious AIM users have convinced others to divulge sensitive information, such as usernames, passwords, and credit card numbers. For example, AIM users have posed as AOL employees and asked to verify credit card information or asked users to verify their AOL screen names and passwords.

File Transfers Reveal IP Address
Engaging in a file transfer, image transfer, voice chat, or file sharing can reveal an AIM user's true IP address. Once an IP address is known, it is possible for a malicious user to concentrate on your system for the purpose of cracking it. Also, it is possible that this information can be used to make the computer a target of a Denial of Service attack.

Theft of Identity
Identity theft can occur due to social engineering (see above) or from a malicious user with the ability to intercept an AIM user's password. There are several utilities that can be used to decrypt AIM passwords, which would enable a malicious user to impersonate another user on AIM. This can also lead to more serious social engineering issues, where a user can mistakenly trust a malicious user and provide sensitive and confidential information.

TECHNICAL COUNTERMEASURES
AIM was designed to be flexible, and is able to work around firewalls and proxies, and can be configured on different ports. Even if traffic for the default AIM ports is blocked, the user has the ability to configure incoming and outgoing TCP sessions on different ports for many of AIM's services.

To prevent just file transfers and file sharing, disable incoming and outgoing TCP sessions on port 5190. This port can be reconfigured via the AIM client to communicate over a different port.

To disable IM images, block incoming and outgoing TCP sessions on port 4443. This port cannot be reconfigured via the AIM client.

To disable AIM completely, access to the host login.oscar.aol.com must be denied on ALL ports. This prevents users from authenticating with an OSCAR server, therefore preventing these users from utilizing all of AIM's services.

These measures will prevent a user from using the AIM service unless a user has access to the web and configures an external proxy server to route Instant Messages. In this situation, a network IDS may be the only way to detect users of this client.

.NET MESSENGER

.NET messenger (formerly MSN Messenger) is the fastest growing Instant Messaging service. Much of this growth is a result of Microsoft shipping this client with Windows XP and the integration of this client with Microsoft Office and Microsoft's Hotmail service.

SECURITY CONCERNS AND .NET MESSENGER

There has been one instance of a propagating worm in .NET Messenger. This particular worm, known as W32/Hello, was not widespread and did very little, if any, damage. The worm relied on users accepting a download (Hello.exe) and manually opening that file. Once opened, this file would send itself to others on the user's Messenger contact list. The method of propagation was actually email, and the worm was so poorly written that there were practically no effects from it.

.NET MESSENGER PROTOCOL

The .NET messenger protocol is an ASCII based protocol, which makes it easy to build clients for different platforms. The protocol is documented by unofficial sources and is available on the Internet.

As in the AIM network, the .NET Messenger network is decentralized. Any server in the .NET Messenger network is able to authenticate clients. Currently, all MSN Messenger servers are in the msgr.hotmail.com sub-domain, contacted via port 1863. The user cannot change this port.

It is possible for users to proxy their .NET messenger connection via the following technologies:

SOCKS 4 Proxy
SOCKS 5 Proxy
HTTP Proxy

.NET Messenger passwords are encrypted using an MD5 hash algorithm. The .NET server generates a seed (a unique string of letters and/or numbers), which is passed to the client in a message such as:

USR 5 MD5 S 989048851.1851137130

To which the client appends the string "Q1P7W2E4J9R8U3S5" to result in a string such as:

989048851.1851137130Q1P7W2E4J9R8U3S5

which is hashed with the password using the MD5 algorithm and returned to the server as a string such as "0212eaad0876afb8505859ca75d21a78".

It is quite difficult to reverse this hash algorithm and retrieve the password from the seed and the hash. However, all messages other than the authentication sequence are in clear text, making it unnecessary for a malicious user to retrieve the password.

.NET MESSENGER CAPABILITIES

.NET Messenger has the following main features:

Instant messaging
Voice/video chat
Application sharing
File transfers
Remote Assistance
Whiteboard

Instant Messaging
Instant messaging is simply the passing of HTML-encoded clear text messages from one user to another, via a central MSGR server. The message is not encrypted and is always routed over the Internet. Messages are sent to the server via TCP port 1863.

Voice/Video Chat
A direct connection must be used for Voice/Video Chat. This data is transferred via a UDP connection to ports 13324 and 13325.

Application Sharing
Application sharing gives a remote user access to programs installed on a computer. Optionally, a user can give control of a program to a remote user. If a user accepts the invitation to share an application, the initiating user may select which of their own programs they wish to share with the other user.

To achieve application sharing, a direct connection is established between clients over the TCP port 1503.

Figure 4. .NET Messenger Application Sharing

File Transfers
File transfers are very similar to image transfers in that a direct connection is established. However, once a file transfer is complete the direct connection is closed.

File transfers in .NET Messenger require a direct connection to be made between two clients. At first, the user who wishes to send a file initiates a request, which passes through a .NET server

and is received by another client. The client that initiated the file transfer listens for the receiver on port 6891/TCP.

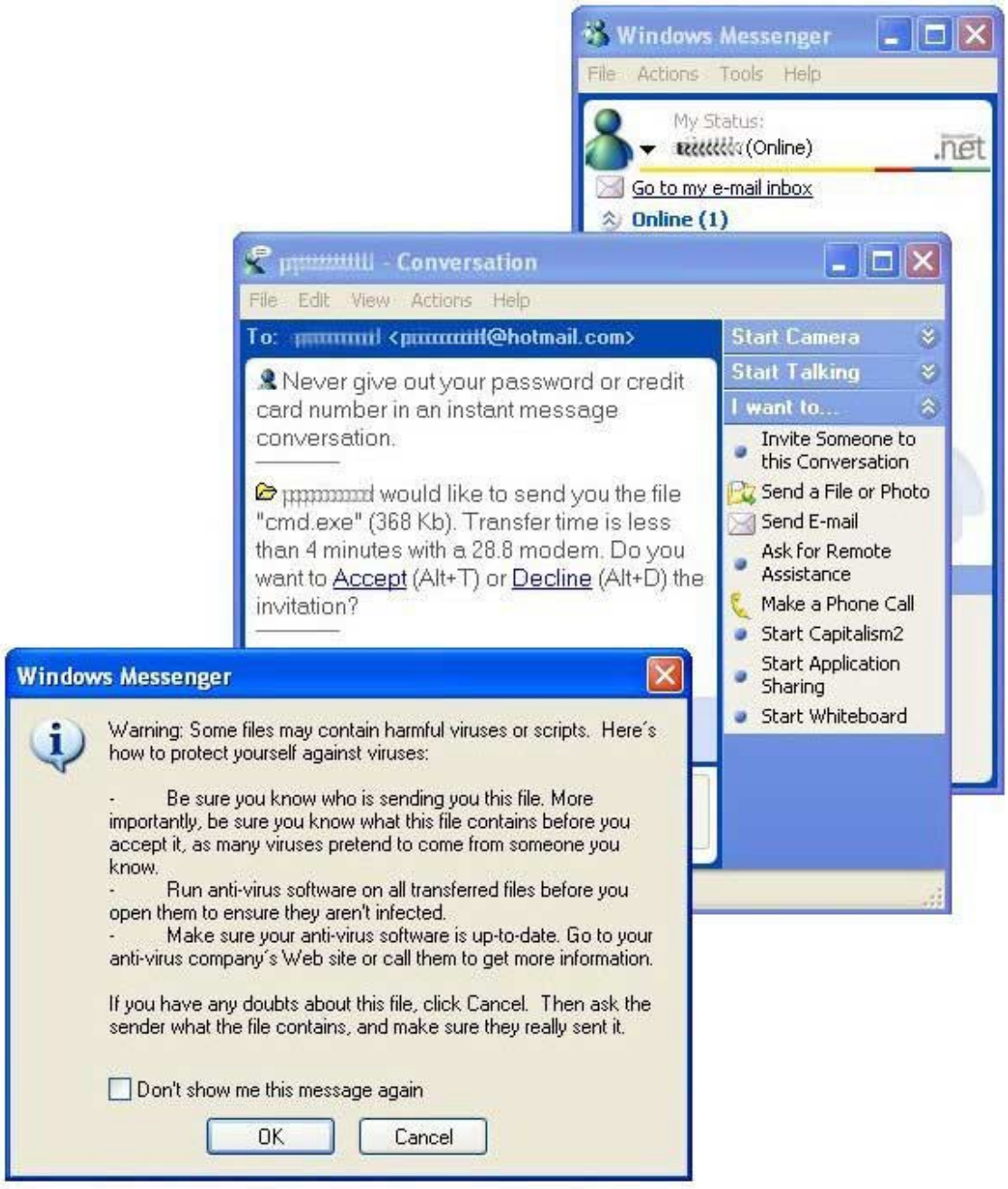

Figure 5. File Transfer Dialog Box in .NET Messenger

Remote Assistance

Windows XP Professional and Home Editions contain the Remote Assistance utility, which allows a remote user to control another computer. The Remote Assistance feature in .NET Messenger launches this utility.

Figure 6. Initiating Remote Assistance in .NET Messenger

Documentation on Remote Assistance in Windows XP is available at:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q300546

Whiteboard

Whiteboard sharing is a way to share a Microsoft Paint document over a direct connection. It is identical to Application Sharing. Starting a whiteboard session with another user is a shortcut of invoking Application Sharer, then selecting Microsoft Paint as the application to share.

Figure 7. Whiteboard Feature of .NET Messenger

.NET SECURITY RISKS

Infected Files: Trojans and Viruses

There have been many reported cases of users being offered files from strangers using .NET Messenger, only to have those files turn out to be a Trojan or virus infected.

Unencrypted Communication

.NET Messenger's protocol stack does not include a secure layer. There is no encryption of any communication sent or received via .NET Messenger. Users may send sensitive data via messages or file transfers. As noted before, messages (and file transfers in most cases) are routed via the Internet. If documents are not strongly encrypted, they should not be sent via .NET Messenger unless they are already a matter of public record. Users should send information over .NET Messenger only with the assumption that a larger audience will have access to this data.

Copyright Infringement

Many file transfers completed through .NET Messenger violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over .NET Messenger eliminates the file size restrictions of email, and, if the recipient is known, .NET Messenger is an easier solution for file transfers compared to FTP.

Social Engineering

Some malicious .NET Messenger users have convinced others to divulge sensitive information such as usernames, passwords, and credit card numbers.

File Transfers Reveal IP address

Engaging in a file transfer, image transfer, voice chat, remote assistance session, or application sharing can reveal a user's true IP address. Once an IP address is known, it is possible for a malicious user to concentrate on your system for the purpose of cracking it. Also, it is possible that this information can be used to make the computer a target of a Denial of Service attack.

Theft of Identity
.NET Messenger's session is based on clear text, making it possible for a malicious user to perform a TCP hijack of an active/idle connection. This malicious user would then be able to impersonate another user in order to obtain sensitive information such as passwords, files, etc.

Technical Countermeasures
.NET Messenger has standard port numbers associated with its features, so it is relatively easy to restrict access to some or all of the program.

To prevent just file transfers, disable incoming and outgoing TCP sessions on port 6891.

To prevent Audio/Video conferencing, block the UDP ports 13324 and 13325.

To prevent Application sharing, block the TCP port 1503.

To disable .NET Messenger completely, deny access to hosts in the msgr.hotmail.com subdomain and block TCP port 1863.

These measures will prevent a user from using the .NET Messenger service unless a user has access to the web and configures an external proxy server to route Instant Messages. In this situation, a network IDS may be the only way to detect users of this client.

YAHOO! MESSENGER

SECURITY CONCERNS AND YAHOO! MESSENGER

Yahoo! Messenger has the weakest security features of the major messaging platforms. Its protocol does not encrypt usernames and passwords, making it risky to even log into the system. Also, the usernames and passwords are sent via HTTP, which allows this information to be stored in HTTP proxy logs.

YAHOO! MESSENGER PROTOCOL

The Yahoo protocol is an ASCII based protocol, and utilizes two ports for its communication. The client sends information via Yahoo! Servers on port 5050 and ASCII information via port 80. Most communication with other clients is handled on port 5050.

Yahoo! Messenger and .NET Messenger's packets share a similar structure. However Yahoo! contains some non-ASCII header information in each packet. When a user signs on to Yahoo! Messenger, the initial authentication packet is sent via HTTP. The username and password are sent in a HTTP 1.0 GET request in clear text. The HTTP server replies with a cookie that is valid for a set amount of time. All further services use this cookie to authenticate.

It is possible for users to proxy their Yahoo connection via any one of the following technologies:

- SOCKS 4 Proxy
- SOCKS 5 Proxy
- HTTP Proxy

YAHOO! MESSENGER CAPABILITIES

Yahoo! Messenger has the following main features:

- Instant messaging
- Voice/video chat
- File transfers
- File sharing

Instant Messaging
Instant messaging is simply the passing of HTML-encoded clear text messages from one user to another, via a central server. The message is not encrypted at all and is routed over the Internet in all cases. Messages are sent to the server via TCP port 5050.

File Transfers
File transfers require a direct connection between peers on port 80/TCP.

A request packet is relayed via the central Yahoo! Server to another user. The sender listens on port 80/TCP for the recipient to accept and connect back, to receive the data. The service uses the common HTTP protocol for file transfers. The end user is able to configure the port that the Yahoo! client listens on for file transfer connections.

File Sharing
File sharing in Yahoo! is a method that allows a user to browse a selected directory structure and to download files. File sharing is optional. However, it is enabled by default.

The connection method for file sharing is the same as for a regular file transfer. The initiator sends a request packet to the target via the Yahoo! server. After the target client accepts the request, the initiator listens on port 80 (this can be changed in the File Sharing options dialog) and the target sends the file list information. Transfers are carried out over the same connection.

The default directory to be shared by Yahoo! Messenger is created by the installation program and is empty by default. This directory can be changed, and if the directory is empty, the connection attempt will fail.

The levels of user security available for File sharing are:

Deny all
Allow from all with Accept File dialog box
Allow from people on “Buddy List” (no dialog box, automatic file transfer)

YAHOO! MESSENGER SECURITY RISKS
Infected Files: Trojans and Viruses
There have been many reported cases of users being offered files from strangers using Yahoo Messenger, only to have those files turn out to be a Trojan or virus infected.

Unencrypted Communication
Yahoo! Messenger’s protocol stack does not include a secure layer. There is no encryption of any communication sent or received via Yahoo! Messenger. Users may send sensitive data via messages or file transfers. As noted before, messages (and file transfers in most cases) are routed via the Internet. If documents are not strongly encrypted, they should not be sent via Yahoo! Messenger unless they are already a matter of public record. Users should send information only with the assumption that a larger audience will have access to this data as well.

Copyright Infringement
Many file transfers completed through Yahoo! Messenger violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over Yahoo! Messenger eliminates the file size restrictions of email, and, if the recipient is known, Yahoo! Messenger is an easier solution for file transfers compared to FTP.

Social Engineering
Some malicious Yahoo! Messenger users have convinced others to divulge sensitive information such as usernames, passwords, and credit card numbers.

File Transfers Reveal IP Address
File transfer or voice chat can reveal a user's true IP address. Once an IP address is known, it is possible for a malicious user to concentrate on your system for the purpose of cracking it. This information may also be used to target the computer in a Denial of Service attack.

Theft of Identity
Yahoo! Messenger's session is based on clear text, making it possible for a malicious user to perform a TCP hijack of an active/idle connection. This malicious user would then be able to impersonate another user to obtain sensitive information such as passwords, files, etc. Furthermore, if the initial sign-on session is captured on the network, or in HTTP proxy logs, a malicious user may simply use the clear text password to login to a users account.

Message Logging
One of Yahoo! Messenger's features records a messaging session to a text file. Malicious access to this information could be used for social engineering or to gain access to sensitive data.

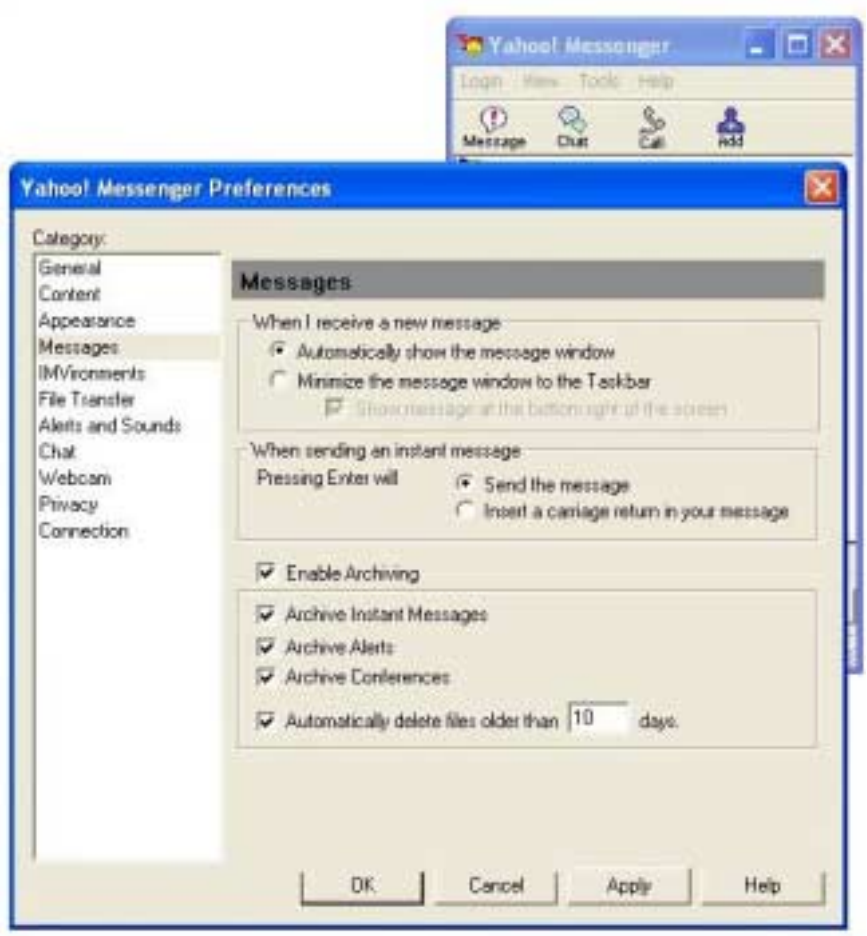

Figure 8. Yahoo! Messenger Preferences and Message Logging

TECHNICAL COUNTERMEASURES
It is somewhat difficult to restrict access to Yahoo! Messenger. Since much of its communication can be routed over port 80, much of the data looks like standard HTTP web traffic.

To prevent Instant messaging, block TCP port 5050.

To disable Yahoo! Messenger completely, deny access to hosts in the *.msg.*.yahoo.com subdomain.

These measures will prevent a user from using the Yahoo! Messenger service unless a user has access to the web and configures an external proxy server to route Instant Messages. In this situation, a network IDS may be the only way to detect users of this client.

ICQ

AOL Time Warner now owns ICQ. However, it currently still maintains a separate database of users from the AIM service.

SECURITY CONCERNS AND ICQ

Programming complexities account for the occasional bug being discovered in the ICQ client. AOL discovered a remote buffer overflow hole in ICQ and released a patch and an advisory on January 24, 2002. There have also been a number of denial of service attacks carried out against the ICQ client.

ICQ PROTOCOL

The ICQ protocol is a binary command based protocol. It has been reverse engineered and is well documented at many Internet sites.

Most of the communication between ICQ users occurs via one or more of the ICQ servers. These servers are reached on port 5190. The login server is known as login.icq.com and while the port number is the same as AIM, the server does not allow the remote user to choose any port as AIM does.

When a user signs on to the ICQ network, their UIN (User Identification Number) is sent along with their password in a packet encrypted with a proprietary algorithm, which has since been reverse engineered.

It is possible for users to proxy their ICQ connection via any one of the following technologies:

SOCKS 4 Proxy
SOCKS 5 Proxy
HTTP Proxy
HTTPS proxy

ICQ CAPABILITIES

ICQ has the following main features:

Instant messaging
Voice/video chat
File transfers
File sharing

Instant Messaging

Instant messaging is simply the passing of HTML clear text messages from one user to another. The message is not encrypted and is always routed over the Internet. Messages are sent via TCP port 3570.

Voice/Video Chat

A direct connection must be used for Voice/Video Chat. This data is transferred via a UDP connection to port 6701.

File Transfers
File transfers require that a direct connection is established. However, once a file transfer is complete the direct connection is closed.

To begin the file transfer, a request packet is sent via the standard Instant Message method to another user. After the user accepts the connection, the receiving client opens a TCP port on 3574 to accept the incoming file.

The remote user must accept all file transfers by clicking Accept in a file transfer request dialog.

File Sharing
File sharing in ICQ is a method that allows a user to browse a selected directory structure and to download files from that directory structure. File sharing is disabled by default. It must be enabled before any sharing can take place.

The connection method is similar to a regular file transfer. The initiator will send a request packet to the target. The target does not need to accept this request as long as the initiator is on their ICQ list. The initiator then connects to the remote system's TCP port 7320 and the target sends the file list information. All file transfers are carried out separate connections that seem to happen on randomly selected TCP ports.

The default directory to be shared is created by the ICQ installation program and may be changed.

Figure 9. ICQ File Sharing

ICQ SECURITY RISKS
The ICQ protocol is known for its lack of strong authentication, lack of encryption, and simplicity. Apart from protocol weaknesses, ICQ has been the target of many DoS bugs and at least one remote buffer overflow.

Infected Files: Trojans and Viruses
There have been many reported cases of users being offered files from strangers using ICQ, only to have those files turn out to be a Trojan or virus infected.

Unencrypted Communication
ICQ does not include a strong encryption model. There is no encryption of any communication sent or received via ICQ. Users may send sensitive data via messages or file transfers. As noted before, messages (and file transfers in most cases) are routed via the Internet. If documents are not strongly encrypted, they should not be sent via ICQ unless they are already a matter of public record. Users should send information over ICQ only with the assumption that a larger audience will have access to this data as well.

Copyright Infringement
Many file transfers completed through ICQ violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over ICQ eliminates the file size restrictions of email, and, if the recipient is known, ICQ is an easier solution for file transfers compared to FTP.

Social Engineering
Some malicious ICQ users have convinced others to divulge sensitive information, such as usernames, passwords, and credit card numbers.

File Transfers Reveal IP Address
Engaging in file transfer or voice chat can reveal a user's true IP address. Once an IP address is known, it is possible for a malicious user to crack the user's system. It is also possible that this information can be used to make the computer a target of a Denial of Service attack.

Theft of Identity
A malicious person with the ability to intercept an ICQ user's password can decipher the password and would then be able to impersonate another user to obtain sensitive information such as passwords, files, etc.

Message Logging
One of ICQ's features is the ability to record a messaging session to a text file. If a malicious user gains access to this file, this information could be used for social engineering or to gain access to sensitive data.

Figure 10. ICQ Message Logging

TECHNICAL COUNTERMEASURES

To prevent access to ICQ from a network, proper access controls are needed.

To prevent standalone file transfers, block TCP sessions on port 3574.

To disable file sharing images, block TCP port 7320.

To disable ICQ completely, deny access to the host login.icq.com on TCP port 5190.

Figure 11. ICQ Connection Preferences and Firewall Options

These measures will prevent a user from using the ICQ service unless a user has access to the web and configures an external proxy server to route Instant Messages. In this situation, a network IDS may be the only way to detect users of this client.

KAZAA/FASTTRACK

KAZAA/FASTTRACK ARCHITECTURE

KaZaA is not strictly a peer-to-peer network, since it still relies on a central server for user information. The central server allows users to sign up for the service, assigns a username and password, is responsible for authenticating users, and assists in locating peers necessary for available downloads.

The server provides a great deal of functionality that is necessary for efficient file sharing and searching within the network. The KaZaA server tracks “SuperNodes”, which are computers that are connected to the network with higher connection speeds. These are automatically chosen by the network to speed up query results. When a peer connects to the server, the server relays the SuperNode’s IP address to the peer. Later, when this peer submits a query, it is dispatched to a nearby SuperNode to find the file. The SuperNode relays requests to the closest computer to begin transfers. This architecture speeds queries; rather than having queries sent to many peers

(as in Gnutella) they are sent to one SuperNode, which significantly reduces traffic generated by searching.

KAZAA FASTTRACK CLIENT

KaZaA (also known as KaZaA Media Desktop) is one of two currently available Fasttrack network clients. KaZaA and Fasttrack are geared mainly towards file sharing and have an architecture and functionality similar to Napster. The other Fasttrack client is called Grokster, and until recently Morpheus connected to this network.

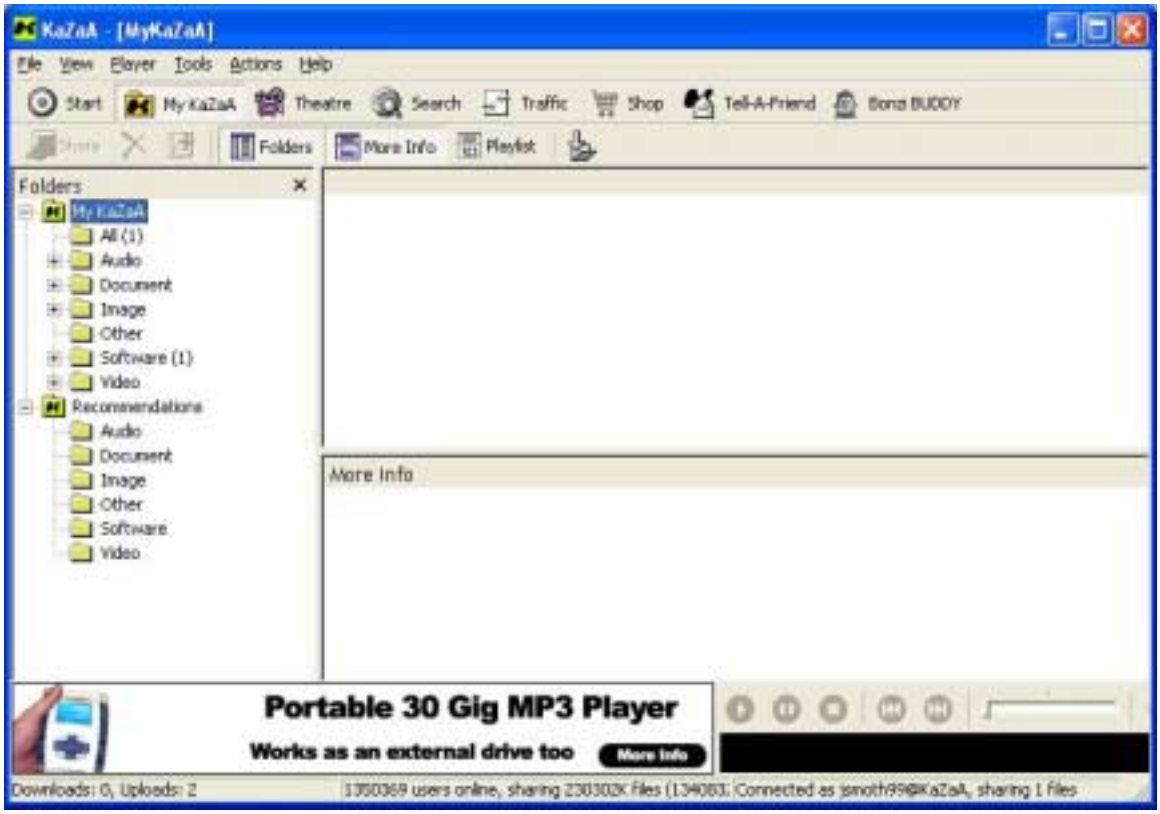

Figure 12. KaZaA Main Screen

SECURITY CONCERNS AND KAZAA

While there haven't been many serious vulnerabilities discovered in this particular client, there are some concerns about the security of the Fasttrack network in general. The network software is closed source and was recently used to maliciously modify the Windows registry of Morpheus users. The client employs a lightweight HTTP server for transferring files, which can be accessed by connecting to http://machineipaddress:1214/

KAZAA (FASTTRACK) PROTOCOL

The Fasttrack protocol is closed source and while it has been reverse engineered to some degree, it changes rapidly to avoid the development of open source clients. Recently, all Fasttrack communication has become encrypted. There appears to be no active projects to reverse engineer the new encryption type.

Encrypted or not, Fasttrack protocol connections are made on TCP port 1214. Clients sharing files listen on port 1214 for incoming connections and clients requesting files make connections outbound to port 1214.

KAZAA CAPABILITIES
In KaZaA, file sharing is enabled by using a default directory created during the installation process. During installation, the user is given the opportunity to select a different directory on their local system to share. Sharing files is not mandatory and it is possible to completely disable the sharing portion of the program.

Once a file is shared, it is accessible to other KaZaA/Fasttrack users to download without authentication.

KaZaA SECURITY RISKS
A denial of service vulnerability was discovered in the KaZaA client and was reported on February 27, 2002. More information is available from:
http://www.securiteam.com/exploits/5RP0R1P6AC.html

KaZaA Bugs
KaZaA client bugs and Fasttrack protocol bugs may exist. The software is closed source and there have already been reports of network-wide security breaches. Recently, the ex-Fasttrack client Morpheus was found to contain a vulnerability that allowed a malicious user to modify registry settings in users Windows registries.
http://www.mp3newswire.net/stories/2002/morpheushack.html

Spyware
All known Fasttrack clients use spyware. Spyware is software that reports back to a vendor site a user's usage habits and patterns. Usually this information is used in an advertising context.

Infected Files: Trojans and Viruses
There have been many reported cases of files downloaded using KaZaA being fraudulent, trojaned or virus infected. Many new viruses are spread this way.

Misconfigured File Sharing
It is possible to misconfigure KaZaA's file sharing feature to include too many, and sometimes private, files. An anonymous KaZaA user might stumble upon sensitive data, such as company documents or system passwords.

Copyright Infringement
Many file transfers completed through KaZaA violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over KaZaA eliminates the file size restrictions of email, and is an easier solution for file transfers compared to FTP.

File Transfers Reveal IP Address
Engaging in a file transfer or file sharing session reveals your true IP address. This may allow someone to concentrate on your system for the purpose of cracking it. It may also make your computer a target of a Denial of Service attack.

TECHNICAL COUNTERMEASURES
To prevent access to KaZaA from a network, proper access controls are needed.

To prevent file transfers and file sharing, block TCP sessions on port 1214.

It must be noted however that if a user has direct access to the web, they may be able to configure an external SOCKS 5 proxy server and route file transfers via this method.

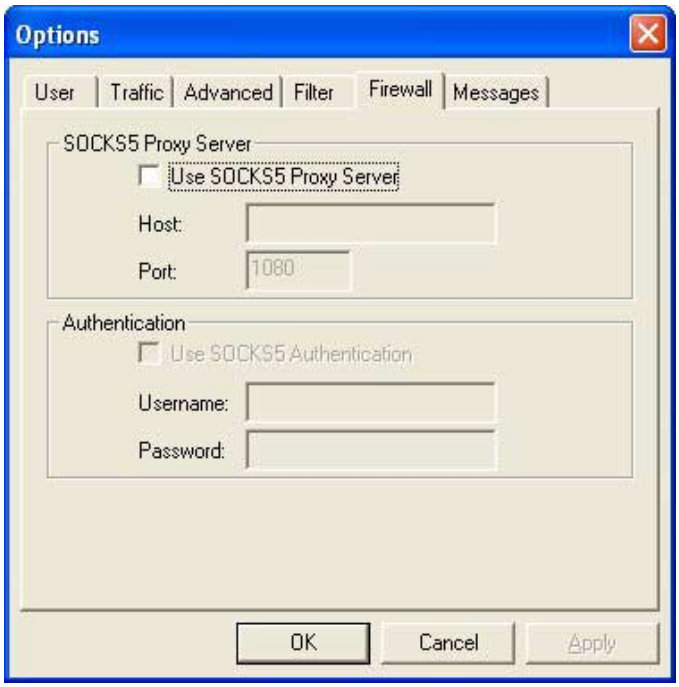

Figure 13. SOCKS 5 Proxy Setup for KaZaA

GNUTELLA

GNUTELLA ARCHITECTURE

The architecture for the Gnutella network is decentralized, and is a true peer-to-peer network involving no central servers for authentication, indexing, etc. To connect to the Gnutella network, it is necessary to connect to several pre-determined IP addresses, which are able to relay information about other servants' IP addresses to a newly connected one. After the servant knows the IP addresses of nearby servants, searching through the network of computers and downloading shared files becomes possible.

When searching for a particular file, the information is first passed to the servants that the originating servant is connected to, and those servants send the query to all the servants they are connected to (and on and on). When downloading a file, the first attempt is made with the servant that reported back. If the download does not work, the request is forwarded from the reporting servant to the other servants it is connected to. The download request is then answered by one of the servants from the reporting servant, and forwarded to the servant that sent the original download request.

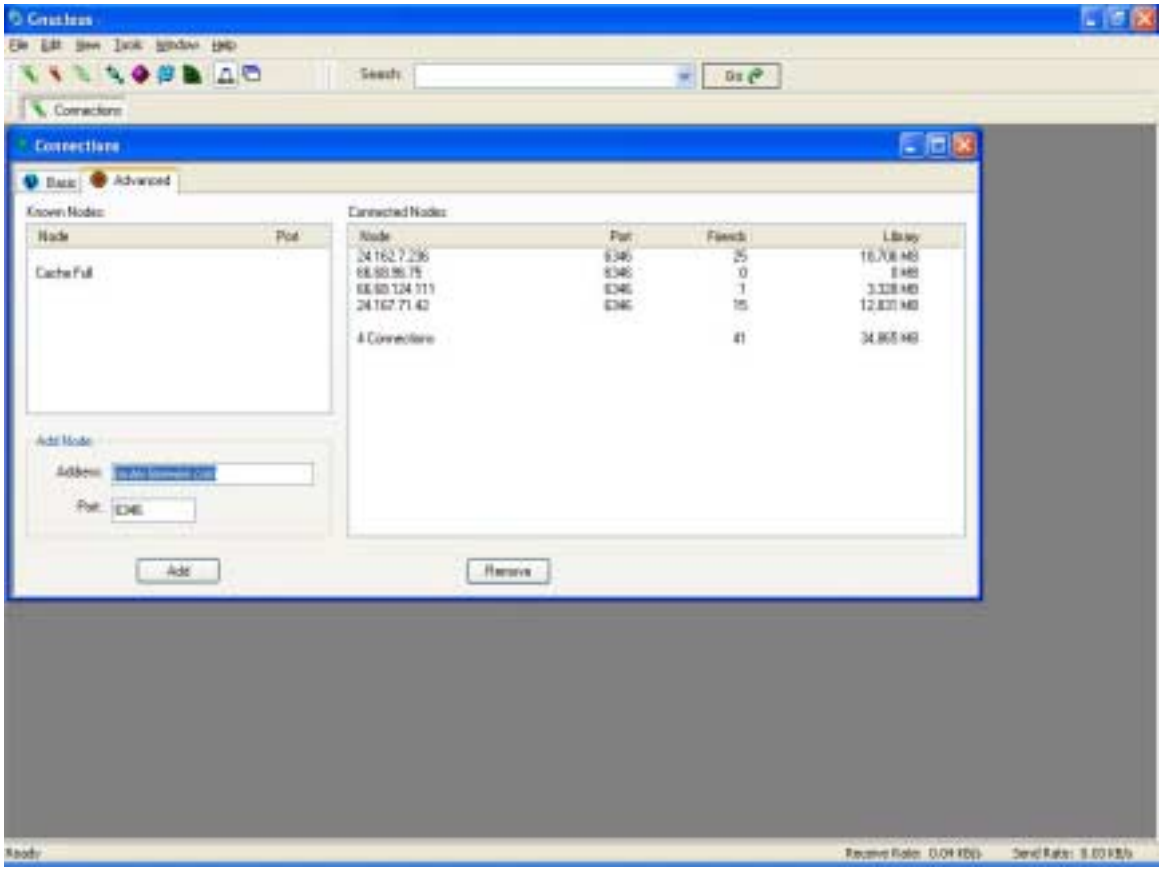

Figure 14. Gnutella Client Gnucleus Connecting to Network

GNUTELLA CLIENTS

Gnutella is a well-published protocol, with many clients that can be used to connect to the network. Morpheus has rewritten its client to utilize the Gnutella network as well, basing it on Gnucleus, an open source client.

Morpheus
http://www.musiccity.com

LimeWire
http://www.limewire.com/

BearShare
http://www.bearshare.com/

Gnucleus
http://www.gnucleus.com

GNUTELLA PROTOCOL

A Gnutella peer sends a ping to another peer to identify the client's presence on the Gnutella network. The ping is forwarded from peer to peer. The ping also includes a TTL (time to live), which limits how long that the ping will be forwarded. Gnutella peers generally set the TTL at 7.

This ping is answered by a pong from the other peer, which includes information such as its IP address and the files that are being shared.

Peers will send the ping to other peers it is connected to. First, it adjusts the TTL so that users who increase their TTL will not flood the network with pings. Also, the client will reduce the TTL by one, sending this ping with a TTL of 6 to all other connected peers. This process of subtracting one from the TTL and then forwarding the ping is continued until the TTL is equal to zero.

Every peer that receives a ping responds to the originating peer with a pong, reversing the route of the ping. These pongs are returned to the client software's hostcatcher, which builds a list of available peers and IP addresses from the collection of this traffic.

When searching for a file, the peer sends a query to every peer listed in the hostcatcher. If a peer finds a matching file, it sends a message to the peer that originated the request, sending the IP address and the filename that matched the query. If a peer does not find a match, it does not respond to the peer that initiated the query.

After a file has been selected for download, the peers communicate via standard HTTP, making it difficult to differentiate the traffic from normal web information.

GNUTELLA CAPABILITIES

Each Gnutella client has different features. However, there are similarities. In Gnutella clients, file sharing is enabled by using a default directory created during the installation process. Some clients allow the user the option to select a different directory on their local system to share. Sharing files is not mandatory. Once a file is shared, it is accessible to other Gnutella network users to download without authentication.

Infected Files: Trojans and Viruses

There have been reported cases of fraudulent, trojaned, or virus-infected files downloaded using Gnutella. Many new viruses are spread this way.

Misconfigured File Sharing

It is possible to misconfigure a Gnutella client's file sharing feature to include too many, and sometimes private, files. An anonymous user might stumble upon sensitive data, such as company documents or system passwords.

Copyright Infringement

Many file transfers completed through the Gnutella network violate copyright laws. It is common for users to send copyrighted software, MP3 files, copyrighted photos, etc. to other users. Trading files over Gnutella eliminates the file size restrictions of email, and is an easier solution for file transfers compared to FTP.

File Transfers Reveal IP Address

Engaging in a file transfer or a file sharing session reveals your true IP address. This information may allow someone to concentrate on your system for the purpose of cracking it. It may also make your system a target of a Denial of Service attack.

TECHNICAL COUNTERMEASURES

Gnutella can use any port for communications on a network, including those that are generally open on a firewall such as port 21, 25, 143, etc. File transfers utilize port 80, therefore even if an administrator were to block Gnutella's default port (6346) both ingoing and outgoing, there is still a method to circumvent security controls. Due to the highly customizable capabilities of Gnutella clients, a network IDS may be the only way to detect users of these clients.

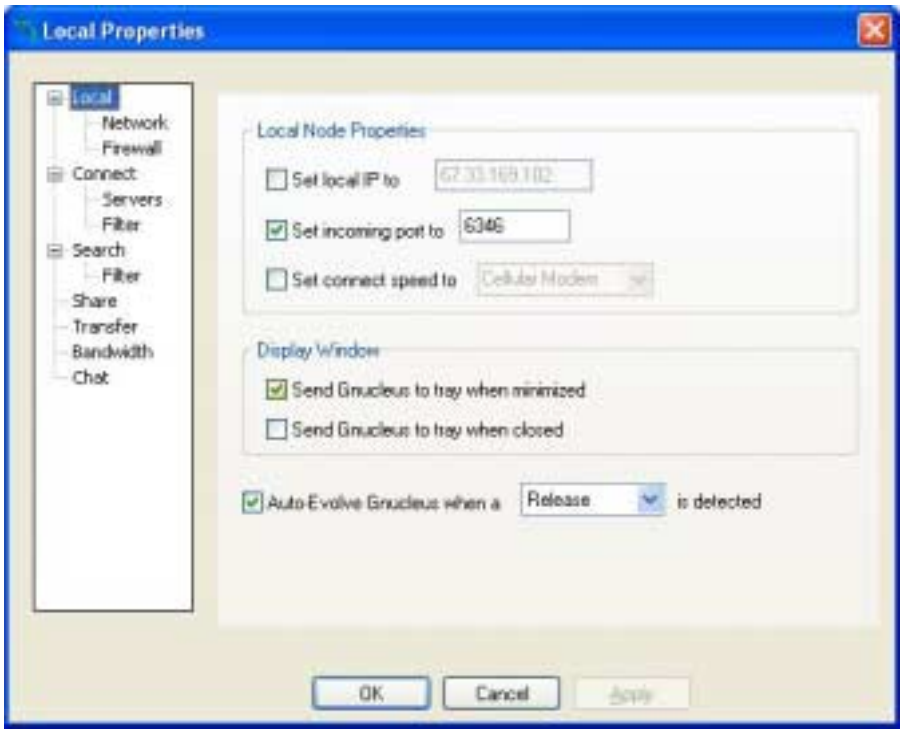

Figure 15. Gnucleus Configuration and Incoming Port Number

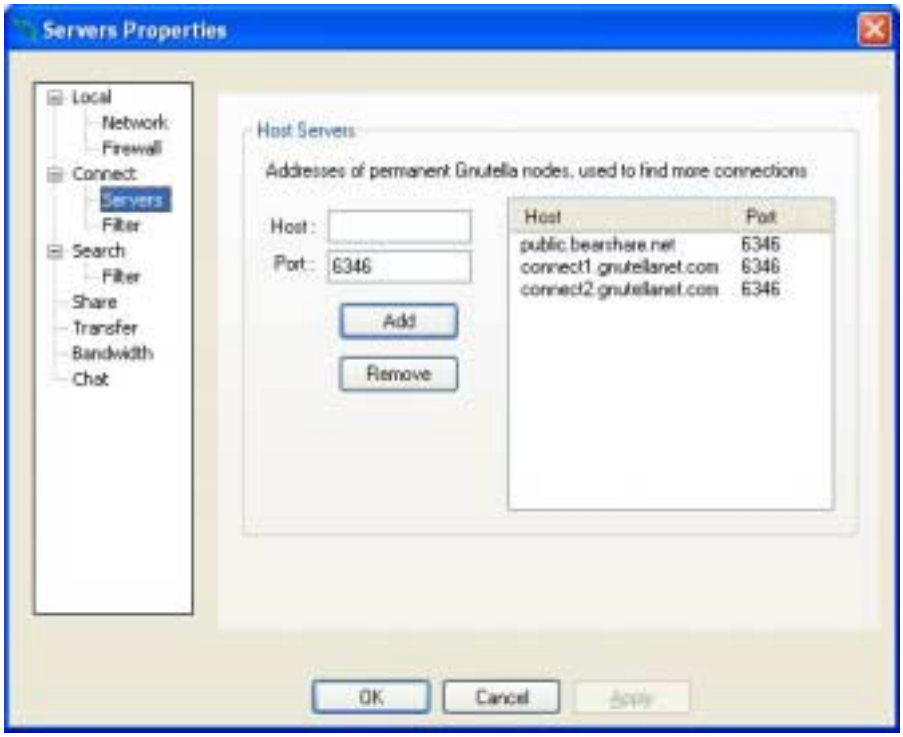

Figure 16. More Gnucleus Configuration and Port Options

INTERNET SECURITY SYSTEMS SOLUTIONS

Internet Security Systems offers a full set of solutions to enable protection against threats that instant messaging and peer-to-peer applications can introduce. Internet Security Systems products and services can play a key role in securing your online assets.

X-FORCE R&D

Internet Security Systems' X-Force knowledge services organization is actively researching instant messaging and peer-to-peer applications to stay ahead of the curve on new threats. This research is used to continually augment products through ***X-Press Update™*** product enhancements with the best defense algorithms and to enhance ISS services offerings.

APPLICATION PROTECTION

Application protection includes file locking (using MD5 checksums), application control, and communications control.

Application Control – Many of today's attacks involve various backdoors. Trojans and worms are commonly delivered via email, web and other Internet technologies. Once your system is infected, these malicious programs can easily be run without your knowledge. Application Protection helps solve this problem by preventing unknown or Trojan infected programs from running without the user's knowledge. You are alerted if a new or altered program tries to run on your computer and you decide whether or not to allow the application to run. Because application protection recognizes new or altered programs on your system, this functionality works without the need for signature updates.

Communications Control – If a malicious or Trojan program manages to infect your computer, it will often attempt to listen or communicate with other computers on the Internet. This may be how the malicious program spreads itself to other systems, or it may be how the program sends your personal information that it has stolen off your computer to a hacker. Communications Control lets you determine which applications are allowed to speak on the Internet, thus preventing unknown or Trojan infected applications from communicating to an attacker

PROTECTION EVOLUTION

ISS has protection solutions to fit your needs at multiple levels. With a strategy of security in-depth, applying protection at the network, server, and desktop can significantly reduce risks and can prevent cyber-harm.

In addition to passive IDS, Internet Security Systems adds active IDS to the protection solution. The solution augments current firewalls and provides the ability to not only detect, but also block attacks from reaching their destination. RealSecure sensors have specific security risk definitions for popular instant messaging programs, providing the ability to detect and/or block peer-to-peer activity. This ability provides an important tool for security administrators to either passively detect activity that may violate company policy, or to actively prevent this activity by blocking file transfers which could not only introduce worms to your network but also could reveal sensitive company information. As a result, intrusion detection evolves into more comprehensive intrusion protection.

Desktop protection plays an important role and is increasingly being recognized as a critical component of a protection strategy, enabling the protection and enforcement of security policy at the desktop level. While personal firewalls help reduce the overall risk, an integrated IDS component allows a user to run applications like NetMeeting and video conferencing software, while protecting and stopping attacks. A traditional personal firewall's option is to block an application if it wants to protect it, while IDS enables the inspection inside the packet for attack analysis.

VULNERABILITY ASSESSMENT
Regular vulnerability scanning is an important part of a proactive security strategy. Internet Security Systems' ***Internet Scanner™*** application provides strong network based assessment capabilities to assist in identifying vulnerabilities and removing them from your network. Internet Scanner includes specific security risk definitions to detect peer-to-peer applications that could introduce risks to your network. The ***System Scanner™*** application assesses host security, detecting and reporting system security weaknesses. The ***Database Scanner™*** application provides application assessment and penetration testing for databases that provide the foundation of enterprise networks and applications. The new ***Wireless Scanner™*** application provides detection and assessment for wireless networking components that expose the network to mobile attackers.

X-FORCE SERVICES
Security is a requirement, but not a core business focus for most organizations. With X-Force Protection Services, an IT administrator adds the collective knowledge and resources of hundreds of trained Internet Security Systems security experts to the company's IT staff. From initial assessments of security posture to on-going management of protection technologies, X-Force Services offer a flexible range of building blocks from which the IT administrator can construct a custom solution. Below are listed just a few of the many services that Internet Security Systems provides.

Assess and Design – Assess your security posture, provide detailed remediation advice, and design a security solution that will guide future security decision-making.

Deploy and Optimize – X-Force professionals can help your IT staff deploy the latest market-leading protection technologies, configuring them for optimal performance in your unique environment.

Manage and Support – X-Force professionals monitor your network and servers for the latest threats, advise you about status, support you in the use of the latest protection technologies, and terminate threats before they become problems.

Educate and Certify – X-Force professionals can train and certify your staff on the latest security technologies, processes, and trends.

Respond and Protect – In the case of a security incident, X-Force professionals can be on site in a matter of hours to help you diagnose the situation, craft a business-appropriate response, and execute the plan.

CONCLUSION
Instant Messaging and peer-to-peer networks are efficient methods for communicating or sharing files online. However, most of these services have minimal security features built into them. Even more dangerous is that several of these applications have the ability to evade corporate security measures. By reconfiguring ports for communication or making traffic appear as HTTP, these applications can escape detection and work through firewalls. The threat of buffer overflows, remote access, and other issues with the client coding are compounded by the threat of malicious users gaining access to sensitive information by social engineering.

Network administrators need to be aware that these services are running and take proper precautions. Risk mitigation includes scanning computers for installed software, as well as installing and configuring protection on the network level. Configuring firewalls to block common ports that these services run on is the first step to prevent malicious activity on the network from affecting computers. A better solution would be to install an IDS, which would be more accurate in detecting the usage of these clients.

APPENDIX

THREAT MATRIX (AS OF APRIL 1, 2002)

Risks	Unencrypted	File Transfer	Known Buffer Overflows	Remote Control	Known Worms	Spyware	Social Engineering	Misuse of Bandwidth/Copyright Issues	Targeted by known viruses
AIM	X	X	X				X	X	X
.NET	X	X		X	X		X	X	X
Yahoo!	X	X					X	X	
ICQ	X	X	X				X	X	
KaZaA	X	X				X		X	X
gnutella	X	X						X	X

About Internet Security Systems (ISS)

Founded in 1994, Internet Security Systems (ISS) (NASDAQ: ISSX) is a pioneer and world leader in software and services that protect corporate and personal information from an ever-changing spectrum of online threats and misuse. Internet Security Systems is headquartered in Atlanta, GA, with additional operations throughout the Americas, Asia, Australia, Europe and the Middle East. For more information, visit the Internet Security Systems Web site at www.iss.net or call 888-901-7477.